



# SEGURETAT EN XARXES I APLICACIONS DISTRIBUÏDES MANET Routing

Memòria del projecte de final de carrera corresponent als estudis d'Enginyeria Superior en Informàtica presentat per Raúl Jiménez Díaz i dirigit per Ramon Martí Escalé.

Bellaterra, Juny de 2009

El firmant, Ramon Martí Escalé, professor del Departament d'Enginyeria de la Informació i de les Comunicacions de la Universitat Autònoma de Barcelona

CERTIFICA:

Que la present memòria ha sigut realitzada sota la seva direcció per Raúl Jiménez Díaz

Bellaterra, Juny de 2009

---

Firmat: Ramon Martí Escalé

*a la meva família*



# Agraïments

Vull donar les gràcies al meu director, Ramon Martí, doncs ha confiat en mi per a la realització d'aquest projecte. La porta del seu despatx sempre ha estat oberta per orientar-me i animar-me. Sempre he sigut benvingut amb el seu somriure i el seu bon humor característic al que ja ens tenia acostumats a les classes. Es dibuixa un somriure a la meua cara quan recordo els seus constructius comentaris a les wikis de l'assignatura Xarxes 1. Per mi ha estat un plaer coneixe'l i compartir amb ell aquests últims anys de carrera.

Als companys amb els que ens ha unit un mateix destí i hem compartit part del nostre viatge.

Als companys que no han pogut continuar aquest viatge perquè han trobat nous reptes a les seves vides.

A Laura Vidal per la seva il·luminació al món del  $\text{\LaTeX}$ , les seves respostes sempre simpàtiques i el seu gran humor.

A David Morcillo per la seva predisposició i col·laboració mútua dels projectes. Per acollir-me al seu despatx i poder realitzar les proves conjuntes.

A Mònica Piñol per els seus consells, la seva paciència i en especial per la seva companyia amb bon humor.

A Miriam perquè des del primer dia d'universitat vam prendre la decisió de recórrer aquest camí junts i mai hauria pogut imaginar trobar una companyia com la seva per fer un viatge com aquest.

A tots un altre cop,

Gràcies.



# Índex

<b>1</b>	<b>Introducció</b>	<b>1</b>
1.1	Motivacions . . . . .	1
1.2	Escenari real . . . . .	2
1.3	Objectius del projecte . . . . .	3
1.4	Contingut de la memòria . . . . .	3
<b>2</b>	<b>Estat de l'art</b>	<b>5</b>
2.1	MANET . . . . .	5
2.2	OLSR . . . . .	7
2.3	MAEMO . . . . .	8
2.4	VMWARE . . . . .	9
<b>3</b>	<b>Anàlisi</b>	<b>11</b>
3.1	Anàlisi de viabilitat . . . . .	11
3.1.1	Viabilitat tècnica . . . . .	11
3.1.2	Viabilitat econòmica . . . . .	12
3.1.3	Viabilitat operativa . . . . .	12
3.1.4	Viabilitat legal . . . . .	13
3.1.5	Alternatives . . . . .	13
3.2	Selecció de protocol . . . . .	14
3.3	Estudi de migració a l'entorn MAEMO . . . . .	14
3.4	Planificació temporal del treball . . . . .	15
3.5	Pressupost . . . . .	15

<b>4 Disseny</b>	<b>17</b>
4.1 Especificacions . . . . .	17
4.2 Selecció millor solució . . . . .	18
4.3 Funcionament intern OLSRD . . . . .	18
4.4 Mòdul OLSRD . . . . .	20
<b>5 Implementació</b>	<b>21</b>
5.1 Preparació de l'entorn de treball . . . . .	21
5.1.1 Construcció Màquina Virtual . . . . .	21
5.1.2 Instal·lació MAEMO development . . . . .	23
5.1.3 Comunicació amb nokia . . . . .	23
5.2 Mòdul OLSRD . . . . .	24
5.3 Instal·lació del plugin a OLSRD . . . . .	25
5.4 Classe JAVA . . . . .	28
5.5 Demostració . . . . .	29
5.5.1 Entorn real . . . . .	29
5.5.2 Entorn virtual . . . . .	32
<b>6 Conclusions</b>	<b>35</b>
6.1 Objectius enlestits . . . . .	35
6.2 Línies de futur . . . . .	36
<b>Bibliografia</b>	<b>37</b>



# Índex de figures

2.1	Llista de xarxes virtuals . . . . .	9
3.1	Planificació temporal del projecte . . . . .	16
4.1	OLSRD: Esquema general del funcionament . . . . .	18
4.2	Plugin OLSRD: Esquema general del funcionament . . . . .	20
5.1	Màquina Virtual per desenvolupar aplicacions MAEMO . . . . .	22
5.2	Espai disponible a la Màquina Virtual . . . . .	23
5.3	Diagrama de la simulació en un entorn real . . . . .	29
5.4	Execució OLSRD al node 169.254.0.1 . . . . .	30
5.5	Conectivitat des del node 169.254.0.1 amb el reste de dispositius .	31
5.6	Esquema de la connexió dels nodes . . . . .	32
5.7	Llista VMware de màquines virtuals . . . . .	33
5.8	Contingut del fitxer /tmp/hosts de la màquina VM0 . . . . .	34
5.9	Comparativa: velocitat en funció del número de salts . . . . .	34

# Capítol 1

## Introducció

### 1.1 Motivacions

Han passat molts anys des de que vaig conèixer per primer cop el departament del DEIC<sup>1</sup>. De totes les assignatures que ofereixen a la nostra titulació, sempre m'agradaven les que tenien relació amb les comunicacions. En el transcurs de les últimes assignatures de xarxes els professors ens anaven introduint en el món dels agents mòbils mitjançant les pràctiques. El tema em va agradar i vaig pensar en la possibilitat de realitzar el PFC sobre quelcom relacionat.

El projecte MANET Routing no toca massa a prop als agents mòbils però sí tracta sobre un tema molt interessant com són les xarxes MANET<sup>2</sup>. Penso que en un futur molt proper, en una societat on la comunicació és bàsica i on la tecnologia és present pràcticament a tot arreu, no serà difícil poder establir xarxes utilitzant com a elements de repetició edificis, cotxes, mòbils o rellotges.

Aquesta llista d'elements no ha sigut aleatòria, he escollit elements de més a menys poder de repetició. Els edificis podrien formar xarxes MESH degut a que aquest tipus de xarxes acostumen a tenir nodes estàtics. Fent servir edificis com a nodes podríem oferir cobertures amb molta fiabilitat a nivell de ciutats.

L'aplicació de xarxes MANET en el món de l'automoció és un tema apassi-

---

<sup>1</sup>DEIC: departament d'Enginyeria de la Informació i de les Comunicacions

<sup>2</sup>MANET, *Mobile Adhoc Networks*

onant doncs aporta un nou ventall de serveis al món de l'automòbil amb un cost zero en tarifes de comunicació. Aquesta tecnologia s'anomena VANET<sup>3</sup> i el seu objectiu principal és proporcionar seguretat i confort als ocupants d'un automòbil. Entre les millores que proposa oferir destaquen avisos de col·lisió, alarmes de senyals de trànsit i fins tot detectar el tràfic que ens envolta per optimitzar la nostra ruta.

Si apliquem aquest concepte als dispositius portàtils ens trobem davant d'una nova era en el món de les comunicacions. Utilitzant Ipv6, que permet 340 sextillons d'adreces, i tenint en compte que gairebé tothom porta alguna mena de dispositiu, podríem convertir-nos en encaminadors humans. Això no és un futur molt esperançador pels operadors telefònics ja que l'usuari no faria servir la xarxa de telefonia. Per això són més reticents a incorporar aquesta tecnologia als telèfons mòbils.

Les xarxes mòbils tenen un futur molt prometedor i aquest projecte és només una primera aplicació d'aquesta tecnologia al servei de les persones.

## 1.2 Escenari real

Ens trobem immersos en una situació d'emergència. Les comunicacions i les estacions d'energia han quedat inutilitzades. Equips de metges i especialistes treballen a peu de carrer intentant arribar a tothom que ho necessiti però veuen limitat el seu treball per una falta de comunicació amb els seus companys.

Aquest seria un escenari on poder tenir una bona xarxa de comunicacions sense disposar d'infraestructura es torna indispensable per tal que els equips a peu de carrer puguin realitzar la seva feina arribant al màxim de persones. Les comunicacions són molt importants i en un escenari crític no podem dependre de la seva disponibilitat. És per això que hem d'assegurar-la amb els propis medis dels que disposem. Per tal d'aconseguir-ho, farem servir les pròpies persones desplegades a peu de carrer per a crear una gran xarxa de comunicació.

---

<sup>3</sup>VANET: Vehicular Ad-Hoc Network

## 1.3 Objectius del projecte

El projecte MANET routing està orientat a buscar una solució per a l'encaminament d'agents mòbils en nodes d'una xarxa MANET<sup>4</sup>. Per aconseguir-ho s'ha dividit en tres fases.

La primera fase consisteix en trobar una implementació d'un protocol d'encaminament per a xarxes MANET. L'equip SENDA ens demana que ha de ser executat en diferents tipus de dispositius com PDA's i ordinadors. Aquesta implementació s'haurà de buscar en sintonia amb les necessitats del projecte global.

Enllestida la primera fase, hem de continuar l'estudi del protocol seleccionat buscant de quins mecanismes disposem per a accedir a la informació d'encaminament. Aquest estudi ens ha de permetre aprendre com extreure un llistat amb totes les direccions accessibles dins la nostra xarxa mòbil. És un pas vital pel projecte, ja que els agents mòbils necessiten disposar d'una llista amb tots els nodes.

Per últim, hauríem d'integrar la llista actualitzada de direccions IP amb els agents mòbils per a que puguin fer ús d'aquesta informació.

## 1.4 Contingut de la memòria

A continuació es presenta un breu resum dels apartats que componen la memòria:

- Estat de l'art: visió teòrica bàsica que ens permetrà conèixer l'estat actual de desenvolupament en que es troben els elements amb els quals es fonamenta el funcionament d'aquest projecte. Parlarem de xarxa MANET, protocol OLSR, sistema operatiu MAEMO i del sistema de virtualització VMware.
- Anàlisi: estudi de viabilitat del projecte en l'àmbit tècnic, econòmic, operatiu i legal, incloent la planificació temporal inicial projecte i el pressupost estimats. A més, veurem els diferents protocols que es poden utilitzar per a la comunicació de nodes dins la xarxa MANET i la viabilitat alhora d'adaptar MAEMO a les PDA Nokia 810.

---

<sup>4</sup>MANET, *Mobile Adhoc Networks*

- Disseny: es defineixen els requeriments del projecte i s'aprofundeix en el funcionament del protocol escollit. Veurem quines possibilitats ens ofereix el mòdul OLSRD.
- Implementació: primerament veurem com s'ha preparat l'entorn de treball, és a dir, les característiques de la màquina virtual que s'ha construït, com instal·lar MAEMO development a les Nokia 810 i com comunicar-nos amb ella. A continuació s'explica com incorporar el plugin OLSRD i la classe JAVA creada. Finalment, es mostren els resultats obtinguts de les proves.
- Conclusions: resum dels resultats obtinguts i proposta de possibles millores.
- Apèndix: s'adjunta amb aquesta memòria un DVD amb els fitxers de codi generats i la màquina virtual per desenvolupar aplicacions MAEMO.

# Capítol 2

## Estat de l'art

En aquest capítol introduïrem l'estat actual dels diferents conceptes que intervenen al projecte MANET ROUTING.

Començarem parlant del tipus de xarxa que hem utilitzat i farem un repàs als protocols que es poden fer servir.

A continuació fixarem la nostra atenció en el protocol seleccionat per al projecte, OLSR. Comentarem les característiques més importants, els seus avantatges i els inconvenients.

Deixarem de banda la temàtica de comunicacions per a parlar del sistema operatiu utilitzat a les PDA's Nokia, MAEMO.

I per finalitzar destacarem el sistema de virtualització VMWARE, utilitzat al projecte en dues ocasions. Primer per a construir una màquina virtual de desenvolupament de software per a MAEMO. En segon lloc, hem tornat a fer servir VMWARE generant un entorn virtualitzat de 10 màquines per a fer una demostració del projecte.

### 2.1 MANET

MANET és un tipus de xarxa Ad-hoc<sup>1</sup> amb la particularitat que els seus nodes estan en moviment. Aquests, es connecten els uns amb els altres facilitant la

---

<sup>1</sup>Ad-hoc, xarxa sense fils on no existeix un node central

mobilitat dels paquets d'informació entre ells. Degut a la naturalesa mòbil dels nodes, la topologia de la xarxa pot anar canviant de manera ràpida i imprevisible.

Per poder fer routing dins d'una xarxa MANET no existeix un protocol ideal. Depenent del context haurem de seleccionar el protocol que per les seves característiques millor s'adapti als requeriments. Podríem classificar aquests protocols en els següents grups:

- Protocols pro-actius: faciliten a cada node una taula amb la llista de hosts actius. Per tal de mantenir-la actualitzada els nodes intercanvien informació regularment. Tot i això, en els intervals entre cada comunicació és possible que la taula no coincideixi amb la realitat.
- Protocols reactius: van aparèixer com a solució al problema dels pro-actius. Busquen millorar la taula d'encaminament de manera que estigui el més actualitzada possible. A canvi generen molt tràfic dins de la xarxa i triguen més temps en mostrar les rutes.
- Protocols orientats al flux de dades: aprofiten el flux de dades per descobrir les noves rutes. Com a característiques podem destacar que en un primer moment son lents investigant els nodes que els envolten.
- Protocols híbrids: Aquest tipus de protocol uneix els avantatges dels pro-actius i els reactius. En conseqüència també hereta els seus problemes. La millora que obtenim amb els protocols adaptables depèn del número de nodes de la xarxa i del volum del flux de dades existent entre els nodes.
- Protocols jeràrquics: En un primer moment s'estableixen les rutes utilitzant un tipus de protocol pro-actiu i a continuació en nivells inferiors utilitza un tipus reactiu. L'eficiència d'aquest protocol va lligada amb la profunditat de la jerarquia.
- Protocols geogràfics: Són protocols que tenen en compte les distàncies físiques dels nodes i la distribució dels nodes a zones importants per al rendiment de la xarxa.

## 2.2 OLSR

Quan parlem d'OLSR<sup>2</sup> ens referim a un protocol d'encaminament IP optimitzat per a xarxes ad-hoc mòbils (MANET).

El protocol OLSR té en compte les característiques d'un entorn hostil i canviant d'una xarxa wireless com són: les pèrdues de paquets puntuals, la freqüent mobilitat dels nodes, possibilitat de cicles o disposar de diferents camins per arribar a un mateix node, baixa senyal o disminució de la velocitat, pèrdua de connexió temporal d'un grup de nodes, etc...

Podem fer servir OLSR a una xarxa ethernet. Aquesta no té les característiques d'una wireless i per aquest motiu les pèrdues de paquets seran mínimes, els nodes no seran mòbils i la topologia és estable. Tot i així, OLSR pot aportar una configuració prèvia pràcticament nul·la facilitant la tasca a un usuari inexpert. Només caldria executar a cada node un programa que inicialitzaria un procés de descobriment dels nodes i es va creant la topologia de l'estat de la xarxa per tal que cada estació de treball sigui conscient del seu entorn.

OLSR és un protocol pro-actiu, això vol dir que constantment va enviant missatges a la xarxa per a conèixer el seu entorn. D'aquesta manera podem mantenir en tot moment una taula de routing actualitzada que ens permet conèixer el camí per a accedir a un node. Això és molt important per a aquest projecte perquè no fa falta cap intenció de comunicació per part nostra per poder conèixer la llista de nodes disponibles. Aquest és un dels nostres objectius principals.

Com a contrapartida podem destacar tres punts: la seguretat, el sistema de distribució de direccions IP i una petita inundació de la xarxa.

Respecte a la seguretat, el protocol OLSR necessita enviar informació de routing als seus veïns i per aquest fet és molt vulnerable a un atac de Denegació de Servei (DoS). Aquest atac podria fer que un node quedés inaccessible i si a més aquest node s'utilitza per a encaminar els paquets a una altra subxarxa podríem perdre un gran nombre de nodes. La manera de corregir aquest punt seria encriptant i xifrant els paquets que porten aquesta informació per tal que no puguin ser alterats per terceres persones. Aquest és un tema amb el que s'està treballant

---

<sup>2</sup>OLSR, *Optimized Link State Routing protocol*



actualment.

Per una altra part, no disposar d'un sistema de distribució de direccions IP impossibilita la configuració zero d'una xarxa utilitzant OLSR. Com a solució es pot fer una conversió utilitzant la MAC, amb ipv4 no és possible generar IP única fent servir la MAC. IPv6 dona solució aquest punt.

I per últim, un punt menys crític és la inundació de xarxa degut a que per defecte cada dos segons s'envien missatges als nodes veïns. Per a evitar aquest problema construïm una ruta MPR<sup>3</sup> on cada node selecciona un conjunt de veïns a un salt que denomina nodes MPR. Aquests nodes són els encarregats de distribuir la informació de control (missatges TC). Amb aquests missatges la resta de nodes poden mantenir actualitzades les seves taules de topologia de la xarxa.

## 2.3 MAEMO

El sistema operatiu MAEMO encara és una mica desconegut però té un gran potencial amagat. Només necessita una petita empenta per part de Nokia i introduir-lo en el món de la telefonia mòbil. De moment Nokia l'utilitza a les PDA's (Nokia les defineix com Internet Table) amb un molt bon resultat demostrat per el *feedback* que ofereixen els usuaris a la pròpia pàgina web de MAEMO.

MAEMO està basat en la distribució de Linux Debian i al voltant seu hi ha tota una plataforma amb molts programadors que desenvolupen aplicacions específiques. Si a més nosaltres som programadors i ja disposem d'aplicacions les podem compilar per MAEMO sense dificultats i utilitzar els nostres propis programes. Això fa molt interessant l'ús de MAEMO al nostre projecte.

Com a primer contacte amb MAEMO recomano visitar la seva plana web<sup>4</sup> on recopilen gran quantitat d'informació pels dispositius que utilitzen MAEMO. És una gran comunitat d'usuaris i programadors amb moltes ganes de donar a conèixer aquest sistema operatiu. Podem trobar seccions de notícies on anuncien quan faran la seva pròxima conferència a més de poder-la descarregar online. Disposen

---

<sup>3</sup>MPR: Multipoint Relaying

<sup>4</sup>MAEMO Web site: <http://maemo.org/>

de manuals i unes 500 aplicacions classificades desenvolupades per ells mateixos.

## 2.4 VMWARE

Hem de preparar un entorn de desenvolupament per a la Nokia 810. Amb l'objectiu de poder treballar des de qualsevol ubicació s'ha pensat en la creació d'una màquina virtual que contingui totes les eines necessàries.

VMware és un dels sistemes de virtualització que porta més temps al mercat. Moltes empreses confien en ells per a mantenir els seus servidors virtualitzats. Tot i que VMware comercialitza els seus serveis, disposa d'una versió gratuïta anomenada VMware Server ( <http://vmware.com/products/server/> ). Nosaltres hem confiat en ells fent servir VMware Server per a la creació de la nostra màquina virtual de desenvolupament a l'entorn MAEMO i també per a generar un entorn de proves. VMware permet no només crear màquines virtuals sinó també xarxes virtuals. D'aquesta manera hem pogut simular 9 xarxes independents per a connectar les nostres màquines de l'entorn de proves. Es pot veure a la figura 2.1

Networks		
Name ▲	VMnet	Type
Bridged	vmnet0	bridged
HostOnly	vmnet1	hostonly
HostOnly (2)	vmnet2	hostonly
HostOnly (3)	vmnet3	hostonly
HostOnly (4)	vmnet4	hostonly
HostOnly (5)	vmnet5	hostonly
HostOnly (6)	vmnet6	hostonly
HostOnly (7)	vmnet7	hostonly
HostOnly (8)	vmnet8	hostonly
HostOnly (9)	vmnet9	hostonly

Figura 2.1: Llista de xarxes virtuals



# Capítol 3

## Anàlisi

En aquest capítol farem un anàlisi del projecte. Començarem fent un estudi de la viabilitat que inclourà la viabilitat tècnica, econòmica, operativa i legal. A més comentarem altres alternatives disponibles. A continuació analitzarem els protocols més adients amb l'objectiu d'escollir el que millor s'adapti a les nostres necessitats.

Tot seguit veurem les opcions que tenim per a migrar la nostra l'aplicació al sistema operatiu MAEMO.

Per a finalitzar analitzarem la planificació del projecte i calcularem el pressupost.

### 3.1 Anàlisi de viabilitat

#### 3.1.1 Viabilitat tècnica

Gràcies al departament del DEIC els projectistes hem pogut fer ús d'un laboratori privat amb màquines connectades en xarxa fent servir GNU/Linux. Disposar d'aquest espai és idoni per a fomentar un ambient de treball i concentració.

És interessant poder fer una simulació real amb una mínima quantitat de nodes formant una xarxa MANET<sup>1</sup>. Simular un escenari només amb NOKIA's encara no

---

<sup>1</sup>MANET, *Mobile Adhoc Networks*

és possible doncs no disposem de suficients dispositius. Això per a aquest projecte no és un problema perquè des del punt de vista de xarxa una PDA o un portàtil són sistemes similars.

A més d'utilitzar portàtils també podem fer simulacions amb VMware utilitzant màquines virtuals. Utilitzar un entorn virtualitzat ens permet crear un bon nombre de nodes fent servir una o dues màquines físiques. Aquest tipus de proves ens servirà per a veure la reacció del protocol davant d'un gran nombre de salts.

Una restricció tècnica important és la d'executar el protocol en diferents sistemes operatius com són MAEMO (per a la PDA NOKIA) o GNU Linux (per als portàtils). Això comporta fer un estudi per a veure com hem de fer la compilació a l'entorn MAEMO.

### 3.1.2 Viabilitat econòmica

Per a la realització del projecte hem fet servir software gratuït: l'ordinador personal porta instal·lada una distribució de Linux Debian, per a la creació de la memòria s'ha fet servir  $\text{\LaTeX}$ , les eines de desenvolupament utilitzades també són gratuïtes.

OLSRD és el programa utilitzat per a la creació de la xarxa MANET. Aquest programa també és de caire gratuït.

Per a les simulacions amb ordinadors hem fet servir portàtils personals i ordinadors de sobretaula. La part més important del projecte és poder comprovar l'execució a una PDA Nokia. El departament ha facilitat la PDA i per això no ha fet falta comprar-ne una.

Hem fet servir VMWARE per a crear màquines virtuals. Aquest és un software comercial però ofereixen productes gratuïts. Hem instal·lat VMWARE Server amb una llicència gratuïta que ens permet la utilització a un ordinador.

### 3.1.3 Viabilitat operativa

La naturalesa mòbil d'aquestes xarxes fa que repetir un escenari concret sigui molt difícil. Aquests acostumen a ser canviants i aleatoris donant lloc a la necessitat de

tenir en compte totes les alternatives possibles. Partint de protocols que ja porten anys treballant en aquest entorn, de ben segur farà que el projecte hereti les seves qualitats d'experiència i fiabilitat.

S'ha pensat en les peticions de l'equip SENDA per a poder interactuar amb el sistema d'agents. Hem trobat la solució per a poder integrar ambdós programes i s'ha executat fent servir les PDA's proporcionades per el departament DEIC. Per aquest motiu la viabilitat operativa ha estat assegurada.

### 3.1.4 Viabilitat legal

Durant el projecte s'ha estudiat el codi font de OLSRD que està sota la llicència BSD. Aquesta llicència és més permissiva que la GPL i permet la utilització del codi en programes comercials. S'ha treballat fent servir GNU/Linux tant al laboratori com al portàtil personal. El programari ha estat sempre sota llicències GPL o LGPL. Per a crear un entorn virtualitzat hem fet servir VMware Server 2.0. La llicència és gratuïta i permet l'execució a un equip amb Linux. Donat que no hem utilitzat de forma directa les dades personals de qualsevol individu i/o empresa, no ha fet falta revisar la llei de protecció de dades. Tots aquest punts fan que es pugui garantir la viabilitat legal.

### 3.1.5 Alternatives

Disposem d'un conjunt ampli d'alternatives dins de les comunicacions entre nodes MANET. Com a protocols a estudiar els dos més importants són: AODV<sup>2</sup> (Ad Hoc On-Demand Vector Routing) i OLSR (Optimized Link State Routing). Els dos tenen característiques diferents que haurem de valorar per a poder ajustar millor a les nostres especificacions.

---

<sup>2</sup>AODV, *Ad Hoc On-Demand Vector Routing*

## 3.2 Selecció de protocol

En un primer pas hem de decidir quin protocol utilitzarem per a la comunicació de la nostra xarxa MANET. La selecció l'hem acotat en decidir si el protocol serà pro-actiu o reactiu. Com a representant del protocol pro-actiu utilitzarem OLSR i per als reactius AODV.

Comparant els protocols podem destacar que els dos han assolit un nivell estable i madur amb el corresponent RFC. OLSR destaca per un bon temps de resposta ja que sempre disposa d'una taula actualitzada amb les rutes que hem de seguir per arribar a una IP destí. En canvi AODV ha de calcular aquesta ruta en el moment que fem una petició d'enviament. AODV per tant, ofereix un estalvi tant en tràfic de xarxa com en energia, fet molt important quan parlem de dispositius portatils.

Ens interessa assegurar el destí amb qualitat abans de fer la petició i això només ens ho ofereix OLSR doncs en tot moment guarda una llista de nodes amb accessibilitat quantificant com és de bó cadascun dels enllaços.

Un altre punt a favor d'utilitzar OLSR al nostre projecte és l'alta mobilitat de les Nokia a peu de carrer, és molt probable que els camins canviïn amb facilitat i en aquests casos AODV no és el més apropiat.

## 3.3 Estudi de migració a l'entorn MAEMO

Una de les nostres restriccions és que la solució trobada pugui ser executada a les PDA Nokia 810. Nokia fa servir MAEMO a les seves PDA llavors ens centrarem en l'estudi d'aquest sistema operatiu.

L'elecció del grup SENDA d'aquestes PDA's s'ha basat en: la gran possibilitat de desenvolupament dins de l'entorn MAEMO, la potència del hardware d'aquests petits dispositius i les múltiples opcions en quant a connectivitat de les que disposen de "serie" sense utilitzar cap adaptador extern.

Les Nokia 810N disposen d'un GPS integrat, connexió WIFI (per a formar la xarxa MANET), connexió Bluetooth, càmera de fotos, una generosa pantalla tàctil per a facilitar la interacció i un teclat complet amagat per a poder escriure

dades amb rapidesa i comoditat.

Per a poder crear aplicacions a l'entorn MAEMO hem de fer servir un paquet de desenvolupament disponible a la pàgina <http://maemo.org/>. Conté totes les llibreries i utilitats necessàries per a construir els fitxer binaris que pot executar MAEMO. Per a poder fer un entorn de programació més portable hem decidit que farem aquesta instal·lació dins d'una màquina virtual creada a mida.

### 3.4 Planificació temporal del treball

A la figura 3.1 podem trobar un diagrama de Gantt amb la planificació temporal del projecte.

### 3.5 Pressupost

Concepte	Preu	Unitats (hores)	Total
Anàlisi	20 €	60	1300 €
Disseny	20 €	60	1200 €
Implementació	20 €	80	1600 €
Documentació / Memòria	20 €	100	2000 €
Total	20 €	300	6000 €

Taula 3.1: Pressupost del projecte





Figura 3.1: Planificació temporal del projecte

# Capítol 4

## Disseny

### 4.1 Especificacions

Per a poder dissenyar una bona solució hem de tenir clars quins requeriments ens demanen. Oferir connectivitat a un conjunt de diferents tipus de dispositius com poden ser PDA's i ordinadors. S'ha de fer sense utilitzar un repetidor únic on poder-se connectar degut a que és possible que algun dispositiu es trobi més allunyat o amb obstacles entre mig.

Els nodes necessiten consultar freqüentment una llista amb les direccions de les màquines disponibles. Hem de proporcionar aquesta llista amb garanties de fiabilitat.

Els agents han de ser capaços d'accedir d'una manera ràpida i àgil a aquesta informació. Hem de trobar un sistema per a poder intercanviar les dades d'encaïment.

Minimitzar la utilització de recursos doncs estem tractant amb dispositius mòbils que tenen limitacions de memòria, processador i energia.

Donat que les xarxes de comunicacions mòbils són encara una tecnologia en desenvolupament hem d'assegurar que la nostra solució sigui compatible amb futures versions, actualitzacions, canvis de sistemes operatius o dispositius.

## 4.2 Selecció millor solució

Al capítol anterior hem anat litzat els diferents protocols que són al nostre abast, arribant a la conclusió que el que més s'ajusta a les nostres necessitats és l' OLSR. De les implementacions existents hem seleccionat OLSRD<sup>1</sup> degut al seu gran número d'actualitzacions. Això ens indica que és un projecte viu i que darrera hi ha molta gent treballant constantment. Com a xifra significativa podem dir que en un període d'un any hem observat buit canvis de versions. Un altre punt a favor és la seva facilitat de portabilitat. OLSRD està escrit en C i el seu consum de CPU és mínim. Això afavoreix que es pugui executar a pràcticament qualsevol sistema com poden ser routers, ordinadors antics, telèfons mòbils i PDA's.

OLSRD té implementat un sistema de plugins. Aquests serveixen per a afegir noves funcionalitats personalitzades de manera independent del codi original. Això compleix una de les nostres restriccions doncs aporta immortalitat al nostre treball fent-lo independent de la versió actual del OLSRD.

La llista de direccions IP la facilitarem mitjançant un fitxer de text. Aquest fitxer el situarem en un directori comú al sistema operatiu, qualsevol aplicació que s'executi a la màquina tindrà accés a ell.

## 4.3 Funcionament intern OLSRD

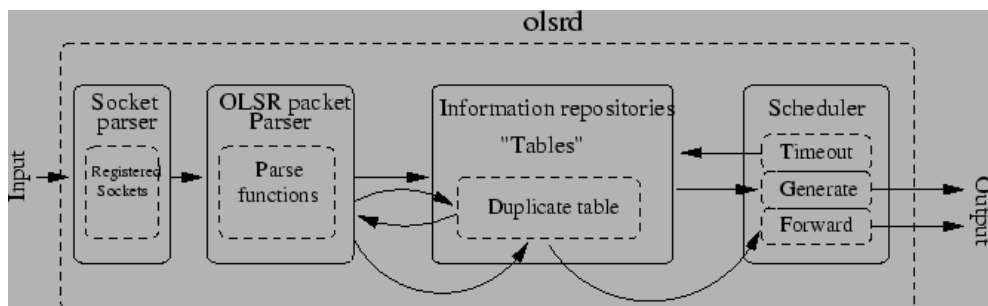


Figura 4.1: OLSRD: Esquema general del funcionament

<sup>1</sup>OLSRD: Optimized Link State Routing Protocol Daemon <http://www.olsr.org>

A la figura número 4.1 podem veure l'esquema general del OLSRD. El primer que apreciem és la seva modularitat. Procedirem a descriure cadascun dels mòduls que el componen:

- *Socket parser*: S'encarrega d'escoltar les dades que ens arriben per la xarxa. Quan arriba un paquet, s'encarrega de cridar a la funció corresponent i enviar-li les dades que han arribat.
- *OLSR packet Parser*: Analitza els paquets de dades que arriben i pren una decisió que pot ser:
  - Descartar el paquet. En cas de que després d'analitzar-lo es consideri no vàlid.
  - Processar el paquet mitjançant unes regles prefixades.
  - Reenviar el paquet segons l'algoritme de reenviament per defecte. Això es fa quan el paquet és vàlid però l'analitzador no coneix aquest tipus de missatges.
- *Information repositories Tables*: Aquest és el cor del OLSRD. Guardem les dades d'encaminament i utilitzem aquesta informació per a fer els càlculs de les rutes o estimacions dels paquets. Les taules enregistren l'estat actual de la xarxa. *Duplicate table* s'utilitza per a mantenir una *cache* dels últims paquets processats. L'actualització d'aquestes taules es fa de manera automàtica mitjançant temps d'espera que elimina les entrades de la taula que no es modifiquen en un cert temps. Cada vegada que succeeix això vol dir que la topologia de la xarxa ha canviat i obliga a recalculer les rutes.
- *Scheduler*: S'encarrega d'executar els events programats amb els intervals de temps corresponents. Aquest mòdul és utilitzat tant internament pel propi OLSRD com pels mòduls programats per l'usuari.

## 4.4 Mòdul OLSRD

Gràcies al bon disseny modular del OLSRD i mitjançant els plugins podem afegir la nostra funcionalitat fent-la independent de la plataforma o versió del programa.

A la figura número 4.2 podem veure com interactuen els plugins amb el OLSRD. És interessant comprovar com mitjançant procediments estandarditzats tenim accés a qualsevol part del OLSRD. Per aquest motiu els mòduls són tan potents. Aprofitant aquesta figura explicarem com funcionarà el nostre plugin.

Farem servir Scheduler per crear un temporitzador que amb un interval de temps prefixat cridi una funció personalitzada. Aquesta funció s'encarregarà de:

- Consultar l'estat actual de les taules del mòdul *Information repositories*.
- Extreure la informació d'encaminament.
- Actualitzar el fitxer de text amb les noves dades.

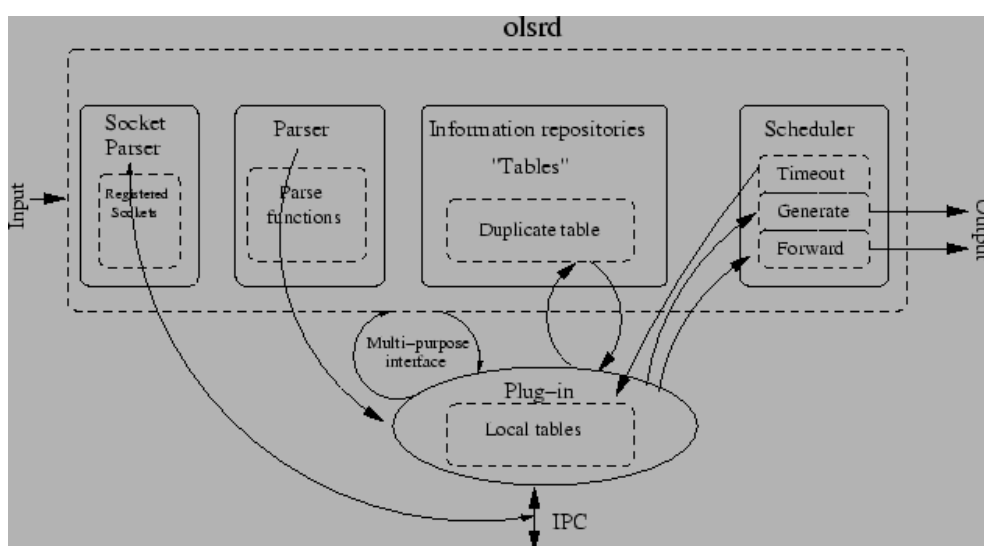


Figura 4.2: Plugin OLSRD: Esquema general del funcionament

# Capítol 5

## Implementació

### 5.1 Preparació de l'entorn de treball

#### 5.1.1 Construcció Màquina Virtual

Hem escollit VMWARE per la seva portabilitat en múltiples plataformes de manera que podem arrencar la nostra màquina en qualsevol sistema operatiu. Amb la intenció de facilitar la distribució d'aquesta MV ens hem fixat com objectiu que el tamany no sobrepassi els 4GB per tal de poder-la adjuntar dins d'un DVD. Les característiques de la màquina virtual les podem veure a la figura 5.1 de les quals podem destacar que la quantitat de memòria RAM escollida ha sigut de 128MB i la capacitat del disc dur total és de 8 GB. S'ha previst poder tenir disponible al voltant del 50% de l'espai en disc per poder-lo utilitzar en els nostres programes.

Una vegada tenim la màquina virtual creada, ens baixem d'Internet una imatge mínima de DEBIAN<sup>1</sup>. Associem aquesta imatge iso al dispositiu de cdrom de la màquina virtual i iniciem l'instal·lació. Instaleu un sistema base mínim sense escriptori ni entorn gràfic. Els paquets addicionals que ens faran falta són:

- open-ssh : Per poder accedir a la nostra màquina via ssh.
- mc : Aquest és opcional, jo l'he instal·lat per comoditat, es un entorn que

---

<sup>1</sup>Debian Netinstall: <http://cdimage.debian.org/debian-cd/5.0.1/i386/iso-cd/debian-501-i386-netinst.iso>

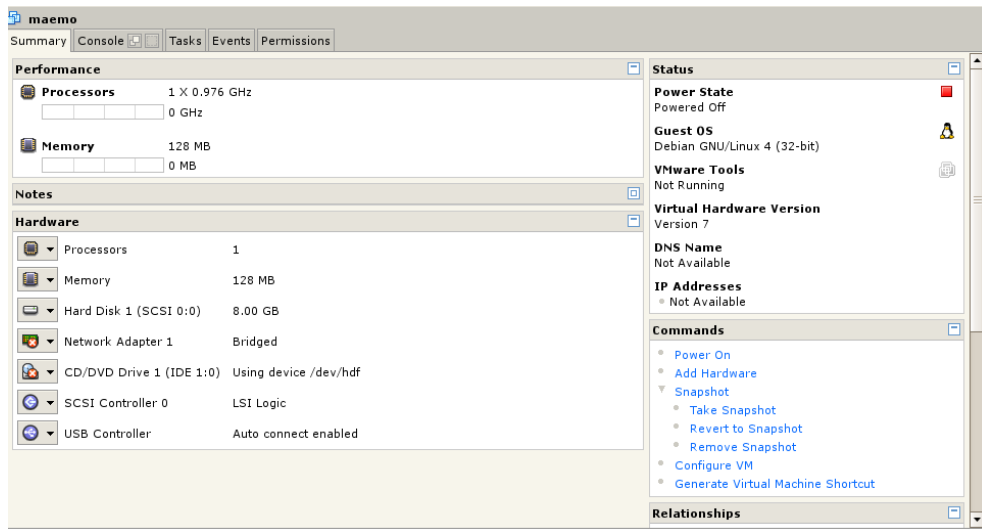


Figura 5.1: Màquina Virtual per desenvolupar aplicacions MAEMO

m'agrada per administrar fitxers.

- **vmtools** : Per a tenir una millor integració amb VMWARE es necessari instal·lat aquest aplicatiu.
- **linux-headers** : Obligatori per a poder compilar aplicacions a la màquina virtual.
- **build-essentials** : Aquest paquet inclou les eines bàsiques de compilació genèriques.
- **psmisc** : Dona access a les utilitats com killall, pstree, fuser, peekfd.
- **iperf**: Per a fer les proves de rendiment de xarxa amb els diferents salts.

En aquest punt ja tenim preparat el sistema per instal·lar les utilitats de MAEMO.

### 5.1.2 Instal·lació MAEMO development

Les instruccions per a instal·lar l'entorn de desenvolupament les podem trobar a la secció de documentació de MAEMO<sup>2</sup>

Quan instal·lem l'entorn de desenvolupament hem de decidir si el volem instal·lar amb Scratchbox. Aquest terme fa referència a la creació d'un entorn protegit i separat de manera lògica del nostre sistema. Normalment és conegut amb el nom Calaix de sorra (Sandbox). És especialment recomanable utilitzar aquesta opció per a compilar aplicacions MAEMO sense preocupar-nos d'afectar a la nostra màquina. En compilar els programes obtenim directament arxius binaris que s'executen a la PDA.

En el nostre cas per a les PDA's de que disposa el departament del DEIC ha fet falta descarregar el Maemo SDK 4.1.2. La instal·lació és molt guiada mitjançant un instal·lador gràfic. La documentació a la que hem fet referència és molt completa amb una gran quantitat de captures de pantalla per a una millor comprensió. Un cop finalitzada l'instal·lació ja disposem de totes les eines per a començar a fer la nostra primera aplicació a MAEMO.

```
maemo:~# df -h
$ ls -l /dev/sda1
Tamaño Usado Disp Uso% Montado en
/dev/sda1 7,6G 4,4G 2,9G 61% /
tmpfs 62M 0 62M 0% /lib/init/rw
udev 10M 80K 10M 1% /dev
tmpfs 62M 0 62M 0% /dev/shm
```

Figura 5.2: Espai disponible a la Màquina Virtual

### 5.1.3 Comunicació amb nokia

Per defecte les PDA Nokia estan protegides per tal de no poder accedir directament al sistema operatiu i que l'usuari interactuï amb les funcions utilitzant la interfície gràfica. No obstant això, existeix una aplicació que obre una pantalla de terminal però amb permisos d'usuari. Si intentem adquirir permisos de *root* (superadministrador) ens trobem amb el problema que no coneixem la contrassenya.

<sup>2</sup>Instal·lació MAEMO SDK: [http://maemo.org/development/training/maemo\\_getting\\_started\\_content/plain\\_html/node4/](http://maemo.org/development/training/maemo_getting_started_content/plain_html/node4/)



Mitjançant la instal·lació d'un paquet anomenat *rootsh* obtenim accés com administradors i d'aquesta manera podem canviar el *password* de l'administrador a una contrassenya coneguda per nosaltres. Amb l'usuari *root* i el seu *password* podem accedir còmodament via SSH des d'un altre ordinador utilitzant una connexió *wireless*.

## 5.2 Mòdul OLSRD

És el moment de presentar el nostre plugin de OLSRD. S'ha escollit com a nom *filew* (abreviatura de file writer) degut a que la seva funcionalitat és escriure un fitxer a disc amb la llista de hosts visibles.

Per a tenir un primer contacte hem utilitzat un mòdul destinat a aquest propòsit. MINI<sup>3</sup> és un plugin integrat a les versions de OLSRD i té una funcionalitat exclusivament didàctica. Ens serveix per a conèixer quina és l'estructura mínima d'un plugin OLSRD. És una de les millors maneres per començar a programar-los.

OLSRD ofereix temporitzadors que es poden crear des de la inicialització dels mòduls. Ens serveix per a cridar les nostres funcions en un instant de temps prefixat. Al codi preparem el nostre temporitzador de la següent manera:

```
/**
 * Initialize plugin
 * Called after all parameters are passed
 */
int olsrd_plugin_init(void)
{
...
olsr_start_timer(3 * MSEC_PER_SEC, 0, OLSR_TIMER_PERIODIC,
                 &new_line, NULL, 0);
...
return 1;
}
```

---

<sup>3</sup>MINI plugin for OLSRD by Bruno Randolph: [bruno.randolf@4g-systems.biz](mailto:bruno.randolf@4g-systems.biz)

Dins la inicialització del plugin cridem a la funció *olsr\_start\_timer*. Mitjançant el primer paràmetre indiquem el període de temps (en mili segons) entre cada execució de la funció indicada al quart paràmetre. En el nostre cas hem escollit tres segons i cridem a la funció *new\_line*.

A la funció *new\_line* trobem:

```
void new_line(void *useless __attribute__((unused)))
{
...
    OLSR_FOR_ALL_RT_ENTRIES(rt) {
        fprintf(sortida,"%s\t%s\n",
            olsr_ip_to_string(&buf1, &rt->rt_dst.prefix),
            get_linkcost_text(rt->rt_best->rtp_metric.cost,
                            true, &lqbuffer));
    } OLSR_FOR_ALL_RT_ENTRIES_END(rt);
...
}
```

Cada vegada que cridem a *new\_line* (per defecte cada tres segons) obrim el fitxer de sortida per escriure i cridem a la funció *OLSR\_FOR\_ALL\_RT\_ENTRIES* passant-li com a paràmetre l'estructura que conté la taula d'encaminament. La funció va recorrent la taula i per cada host envia al fitxer una nova línia amb la ip del host destí i el seu *ETX*<sup>4</sup> associat.

## 5.3 Instal·lació del plugin a OLSRD

En primer lloc explicarem com descarregar i instal·lar l'última versió de l'OLSRD des de la pàgina oficial<sup>5</sup>.

---

<sup>4</sup>Expected Transmission Count: OLSRD l'utilitza per saber quants intents teòrics necessita per arribar a un host

<sup>5</sup>Pàgina oficial OLSRD: <http://www.olsr.org/>

1. Per a poder satisfer els requeriments instal·larem les dependències del paquet OLSRD:

```
# apt-get build-dep olsrd
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes NUEVOS:
  binutils bison build-essential bzip2 cpp cpp-4.3 debhelper diffstat
  flex g++ g++-4.3 gcc gcc-4.3 gettext html2text intltool-debian
  libcompress-raw-zlib-perl libcompress-zlib-perl
  libdigest-sha1-perl libfile-remove-perl libgmp3c2 libgomp1
  libio-compress-base-perl libio-compress-zlib-perl
  libmail-box-perl libmail-sendmail-perl libmailtools-perl
  libmpfr1ldbl libobject-realize-later-perl libstdc++6-4.3-dev
  libsys-hostname-long-perl libtimedate-perl liburi-perl
  linux-libc-dev make po-debconf quilt
0 actualizados, 43 se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 12,6MB/25,8MB de archivos.
```

2. Ens baixem l'última versió (actualment 0.5.6-r4), ho podem fer mitjançant la comanda *wget*:

```
wget http://www.olsr.org/releases/0.5/olsrd-0.5.6-r4.tar.bz2
```

3. Descomprimim el fitxer:

```
tar xjf olsrd-0.5.6-r4.tar.bz2
```

4. Això ens crea un directori anomenat: *olsrd-0.5.6-r4*:

```
cd olsrd-0.5.6-r4
```

5. Per a compilar farem servir la comanda *make*, si tot va bé podem continuar amb la instal·lació fent servir *make install*. És important recordar que l'exe-

cució del *make install* s'ha de fer amb permisos d'administració (*sudo make install*):

```
make && sudo make install
```

6. En aquest moment tenim instal·lat l'OLSRD, ara procedirem amb la instal·lació del plugin. Hem de copiar el fitxer *filew.zip* dins del directori *olsrd-0.5.6-r4/lib* i fer la descompressió:

```
cd olsrd-0.5.6-r4/lib
unzip filew.zip
Archive: filew.zip
  creating: filew/
  inflating: filew/README_FILEW
  inflating: filew/version-script.txt
  inflating: filew/Makefile
   creating: filew/src/
  inflating: filew/src/olsrd_plugin.c
  inflating: filew/src/filew.c
  inflating: filew/src/olsrd_plugin.h
  inflating: filew/src/filew.h
```

7. Per a compilar el plugin primer hem de fer un *make clean* (serveix per netejar possibles restes de compilacions anteriors). Tot seguit ja podem executar la comanda *make* i *make install* (aquesta última també s'ha de fer amb permisos d'administració) (*sudo make install*):

```
# cd filew
debian:~/olsrd-0.5.6-r4/lib/filew# make clean
debian:~/olsrd-0.5.6-r4/lib/filew# make && sudo make install
debian:~/olsrd-0.5.6-r4/lib/filew# ls -l
total 24
-rwxrwxrwx 1 root root 2239 jun 12 01:01 Makefile
-rwxr-xr-x 1 root root 4504 jun 15 23:19 olsrd_filew.so.0.2
```

```

-rwxrwxrwx 1 root root 752 jun 12 01:02 README_FILEW
drwxr-xr-x 2 root root 4096 jun 15 23:19 src
-rwxrwxrwx 1 root root 133 feb 1 20:54 version-script.txt
debian:~/olsrd-0.5.6-r4/lib/filew#

```

8. Si tot va bé, obtenim un fitxer binari anomenat *olsrd\_filew.so.0.2*. El següent pas és modificar l'arxiu de configuració del OLSRD ( */etc/olsrd.conf* ) afegint una línia per a carregar el mòdul. Ho podem fer mitjançant la comanda:

```

echo LoadPlugin \"olsrd_filew.so.0.2\"{ } >> /etc/olsrd.conf

```

9. Per a veure que tot funciona correctament executarem el programa OLSRD (amb permisos d'administrador) i veurem com es crea el fitxer */tmp/hosts*. Per a poder apreciar els canvis podem fer servir la comanda:

```

watch -n1 cat /tmp/hosts

```

## 5.4 Classe JAVA

Per a poder fer una simulació amb col·laboració amb un altre projecte del mateix departament s'ha desenvolupat en cooperació amb David Morcillo (projecte [Mor09]) una petita classe JAVA per a poder accedir desde la seva plataforma a la llista d'adreces IP.

```

public ArrayList onFindNeighbors() {
    ArrayList neighbors = new ArrayList();

    try {
        BufferedReader in =
new BufferedReader(new FileReader("/tmp/hosts"));
        String line;
        while ((line = in.readLine()) != null)

```

```
neighbors.add(line);  
} catch (FileNotFoundException ex) {  
  
} catch (java.io.IOException ex) {  
  
}  
  
return neighbors;  
}
```

El funcionament és trivial doncs el que fem és recórrer el fitxer i anar omplint una variable amb la llista d'IPs.

## 5.5 Demostració

### 5.5.1 Entorn real

El nostre objectiu per a aquesta simulació és comprovar l'execució del OLSRD a la Nokia i assegurar la seva integració amb altres dispositius de la xarxa.

Per a fer aquesta simulació hem utilitzat tres portàtils i una Nokia 810. El nostre objectiu és construir una xarxa com la figura 5.3.

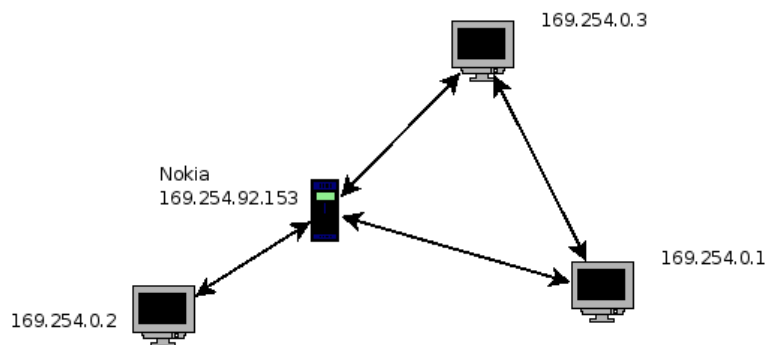


Figura 5.3: Diagrama de la simulació en un entorn real

Per a aconseguir-ho ens ha fet falta aïllar les tarjetes inalàmbriques mitjançant paper d'alumini. La nostra intenció és que la PDA fes de pont i els paquets poguessin saltar a través de la Nokia per a accedir als diferents nodes.

```

*** olsr.org - 0.5.6-r3 (2008-12-26 15:49:51 on acer) ***

--- 09:50:09.125216 ----- LINKS

IP address      hyst      LQ      ETX
169.254.0.3      0.000    1.000/0.992    1.008
169.254.92.153   0.000    1.000/1.000    1.000

--- 09:50:09.125295 ----- TWO-HOP NEIGHBORS

IP addr (2-hop)  IP addr (1-hop)  Total cost
169.254.0.2      169.254.92.153   2.232

--- 09:50:09.125335 ----- TOPOLOGY

Source IP addr  Dest IP addr      LQ      ETX
169.254.0.1     169.254.0.2       0.392/0.867    2.942
169.254.0.1     169.254.0.3       1.000/0.988    1.012
169.254.0.1     169.254.92.153    1.000/1.000    1.000
169.254.0.2     169.254.92.153    0.898/0.914    1.219
169.254.0.3     169.254.0.1       0.992/1.000    1.008
169.254.0.3     169.254.92.153    0.863/0.247    4.692
169.254.92.153  169.254.0.1       1.000/1.000    1.000
169.254.92.153  169.254.0.2       1.000/0.898    1.114

```

Figura 5.4: Execució OLSRD al node 169.254.0.1

A la figura 5.4 podem veure l'execució del OLSRD fent servir l'opció *DEBUG=1*. Ens ofereix informació actualitzada de l'estat actual de la topologia. En aquest exemple observem com des del node 169.254.0.1 tenim accés directe amb la Nokia (169.254.92.153) i un altre portàtil (169.254.0.3). El valor *ETX* ens està indicant l'estimació (basada en el valor  $LQ^6$ ) del número teòric de reintents per a arribar a aquest node. Una mica més aball trobem que tenim access al node 169.254.0.2 mitjançant dos salts i fent servir la Nokia (169.254.92.153) com intermediari.

A la figura 5.5 podem veure una prova de connectivitat amb la Nokia i amb el portàtil que hi ha darrera de la Nokia. El resultat d'aquest tests ha sigut totalment satisfactori.

---

<sup>6</sup>LQ: Link Quality

```
64 bytes from 169.254.92.153: icmp_seq=35850 ttl=64 time=4.63 ms
64 bytes from 169.254.92.153: icmp_seq=35851 ttl=64 time=5.74 ms
64 bytes from 169.254.92.153: icmp_seq=35852 ttl=64 time=37.5 ms
64 bytes from 169.254.92.153: icmp_seq=35853 ttl=64 time=2.02 ms
64 bytes from 169.254.92.153: icmp_seq=35854 ttl=64 time=2.15 ms
64 bytes from 169.254.92.153: icmp_seq=35855 ttl=64 time=1.99 ms
64 bytes from 169.254.92.153: icmp_seq=35856 ttl=64 time=2.00 ms
64 bytes from 169.254.92.153: icmp_seq=35857 ttl=64 time=2.11 ms
64 bytes from 169.254.92.153: icmp_seq=35858 ttl=64 time=2.65 ms
64 bytes from 169.254.92.153: icmp_seq=35859 ttl=64 time=2.37 ms
64 bytes from 169.254.92.153: icmp_seq=35860 ttl=64 time=1.95 ms
64 bytes from 169.254.92.153: icmp_seq=35861 ttl=64 time=22.9 ms
64 bytes from 169.254.92.153: icmp_seq=35862 ttl=64 time=1.84 ms
64 bytes from 169.254.92.153: icmp_seq=35863 ttl=64 time=1.85 ms
64 bytes from 169.254.92.153: icmp_seq=35864 ttl=64 time=6.69 ms
64 bytes from 169.254.92.153: icmp_seq=35865 ttl=64 time=2.07 ms
^C
--- 169.254.92.153 ping statistics ---
35865 packets transmitted, 35194 received, +420 errors, 1% packet loss, time 35933876ms
rtt min/avg/max/mdev = 1.335/4.342/2033.388/17.266 ms, pipe 4
acer:~# ping 169.254.0.2
PING 169.254.0.2 (169.254.0.2) 56(84) bytes of data.
64 bytes from 169.254.0.2: icmp_seq=1 ttl=63 time=8.06 ms
64 bytes from 169.254.0.2: icmp_seq=2 ttl=63 time=5.21 ms
64 bytes from 169.254.0.2: icmp_seq=3 ttl=63 time=16.1 ms
64 bytes from 169.254.0.2: icmp_seq=4 ttl=63 time=5.33 ms
64 bytes from 169.254.0.2: icmp_seq=5 ttl=63 time=5.29 ms
```

Figura 5.5: Conectivitat des del node 169.254.0.1 amb el reste de dispositius



### 5.5.2 Entorn virtual

Per poder comprovar la funcionalitat del plugin a un entorn amb una gran quantitat de subxarxes i salts hem preparat l'escenari de la figura 5.6.

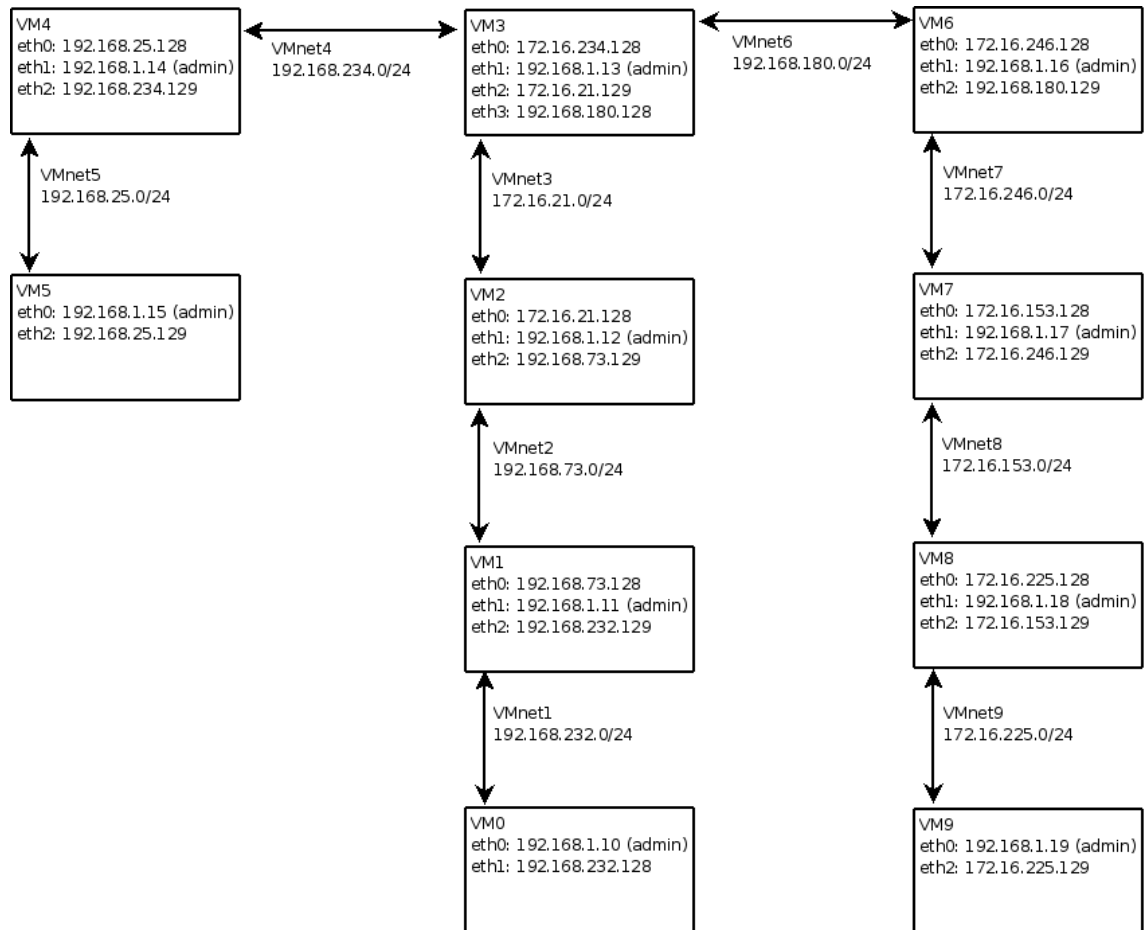
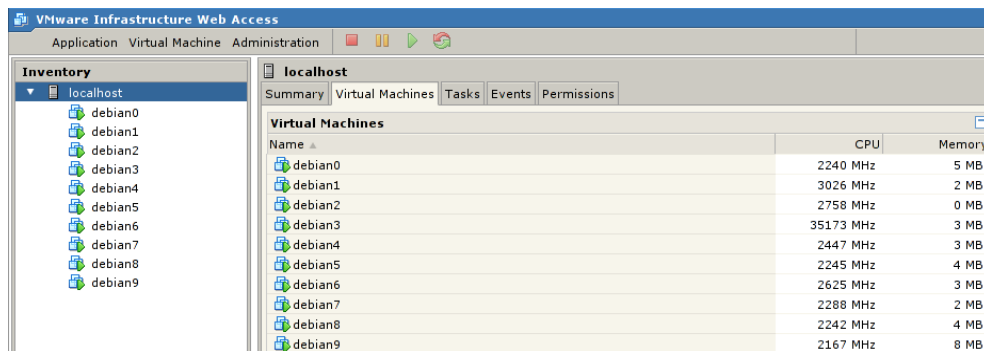


Figura 5.6: Esquema de la connexió dels nodes

Per aïllar les màquines (figura 5.7) hem creat nou xarxes virtuals que les connectaran d'una en una.

Totes les màquines virtuals disposen d'una interfície de xarxa que hem anomenat *admin* i l'utilitzem per administrar el node. Aquesta interfície no intervé en la simulació OLSR i podem dir que forma part d'una xarxa paral·lela independent.

Podem trobar 3 tipus de màquines:



Name	CPU	Memory
debian0	2240 MHz	5 MB
debian1	3026 MHz	2 MB
debian2	2758 MHz	0 MB
debian3	35173 MHz	3 MB
debian4	2447 MHz	3 MB
debian5	2245 MHz	4 MB
debian6	2625 MHz	3 MB
debian7	2288 MHz	2 MB
debian8	2242 MHz	4 MB
debian9	2167 MHz	8 MB

Figura 5.7: Llista VMware de màquines virtuals

- Màquina fulla: és una màquina que no pot fer l'encaminament de paquets doncs només té una interfície de xarxa (sense incloure la d'administració *admin*). Les trobem al final d'un camí. Les màquines VM5, VM9 i VM0 són d'aquest tipus.
- Màquina bifurcació: al exemple només trobem una màquina d'aquest tipus. És la VM3 i serveix per connectar 3 subxarxes. Disposa de tres interfícies de xarxa sense tenir en compte la d'administració *admin*.
- Màquina de pas: les anomenem així perquè els paquets acostumen a passar a través d'elles. Simulen un salt i disposen de dos interfícies de xarxa.

Executem OLSRD a totes les màquines i després d'uns segons consultem el fitxer `/tmp/hosts` com podem veure a la figura 5.8.

Podem observar com el valor ETX situat a la dreta de l'adreça IP ens aporta una ràpida visió de la xarxa. Gràcies a ell podem intuir quines són les màquines més llunyanes des de la nostra posició.

Aprofitant la simulació hem instal·lat el programa *Iperf* a tots els nodes per poder extreure uns valors de rendiment numèrics en funció del número de salts. El resultat obtingut el podem trobar a la figura 5.9

```

cocho@cochostation:~$ cat /tmp/hosts
172.16.21.128 2.090
172.16.21.129 3.161
172.16.153.128 5.202
172.16.153.129 6.218
172.16.225.128 6.218
172.16.225.129 7.230
172.16.246.128 4.161
172.16.246.129 5.202
192.168.25.128 4.161
192.168.25.129 5.256
192.168.73.128 1.000
192.168.73.129 2.090
192.168.180.128 3.161
192.168.180.129 4.161
192.168.232.128 1.000
192.168.232.129 1.000
192.168.234.128 3.161
192.168.234.129 4.161
cocho@cochostation:~$ 

```

Figura 5.8: Contingut del fitxer /tmp/hosts de la màquina VM0

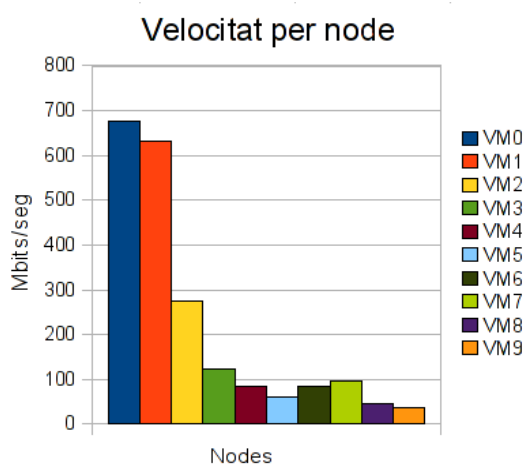


Figura 5.9: Comparativa: velocitat en funció del número de salts

# Capítol 6

## Conclusions

### 6.1 Objectius enllestits

El projecte MANET Routing el podem dividir en 2 objectius principals.

- Crear una xarxa MANET utilitzant les PDA's en un entorn mòbil. Després d'analitzar els diferents protocols hem trobat una solució a mida pensant en els requeriments del projecte. Es va donar una solució a l'execució del protocol al sistema operatiu de Nokia fent servir una màquina virtual amb el paquet de desenvolupament MAEMO. Per a comprovar aquesta solució hem fet proves (figura 5.3) obtenint resultats satisfactoris.
- Poder consultar una llista d'adreces IP visibles des del node actual. Per donar solució a aquest punt hem creat un plugin per l'OLSRD. Fent servir un plugin assegurem la immortalitat de la nostra feina doncs el nostre treball no depèn d'una versió concreta del programa.

Periòdicament els projectistes de l'equip SENDA ens reuníem per intercanviar opinions. En una d'aquestes reunions va sorgir una proposta per millorar el plugin. Volien afegir un indicador a la llista d'adreces IP que mostrés la qualitat de cada node. Va ser per aquest motiu que es va començar a treballar en la versió 2 del plugin. Aquesta versió és completament operativa com es pot veure a la figura 5.8

## 6.2 Línies de futur

Encara que hem assolit tots els objectius existeixen un parell de punts a millorar.

- El direccionament automàtic. Actualment el sistema d'agents necessita per poder saltar de plataforma el nom del host destí. Per aquest motiu tots els nodes comparteixen un mateix fitxer */etc/hosts* i disposen d'una adreça fixa. De ben segur en un futur serà possible fer el salt només amb la direcció IP. Per donar una solució proposo fer servir IPv6 i associar una adreça al node fent servir la seva MAC.
- La seguretat del protocol. OLSR és un protocol en desenvolupament i necessita reforçar la seva seguretat. Hauríem de millorar aquest aspecte fent servir connexions xifrades per a l'intercanvi d'informació de control entre els nodes garantint la integritat del paquet.

# Bibliografia

- [Mor09] Morcillo Muñoz, David. Agents mòbils en situacions d'emergencia. Gestió dinàmica de serveis. Projecte Final de Carrera. Bellaterra: Universitat Autònoma de Barcelona, 2009.
- [med06] Comparativa de los protocolos AODV y OLSR con un emulador de redes Ad-Hoc, febrer 2006.  
Alejandro Medina Santos, Mari Carmen Domingo Aladrén
- [inmic2001] Optimized Link State Routing Protocol, 2001.  
T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, a. Qayyum et L. Viennot, IEEE INMIC Pakistan 2001  
<<http://hipercom.inria.fr/olsr/inmic2001.ps>>
- [medhoc] Simulation Results of the OLSR Routing Protocol for Wireless Network, 2002.  
A. Laouiti, P. Muhlethaler, A. Najid, E. Plakoo, 1st Mediterranean Ad-Hoc Networks workshop (Med-Hoc-Net), Sardegna, Italy 2002  
<<http://hipercom.inria.fr/olsr/medhoc.ps>>
- [olsr] Pàgina principal de l'OLSR, Juny 2009 .  
<<http://www.olsr.org>>

---

Firmat: Raúl Jiménez Díaz  
Bellaterra, Juny de 2009

## **Resum**

L'objectiu del projecte és tractar de donar solució al problema d'interconnexió de dispositius mòbils, en concret de PDAs Nokia 810, en una situació on no es disposa d'un punt d'accès fixe. Per a fer-ho possible es forma una xarxa MANET, és a dir una xarxa Ad-hoc mòbil, on els nodes són els propis dispositius. Un cop establerta la connexió, s'obté un llistat actualitzat amb les adreces IP de tots els nodes visibles de la xarxa.

## **Resumen**

El objetivo del proyecto es tratar de dar solución al problema de interconexión de dispositivos móviles, en concreto de PDAs Nokia 810, en una situación donde no se dispone de un punto de acceso fijo. Para ello se forma una red MANET, es decir una red Ad-hoc móvil, donde los nodos son los propios dispositivos. Una vez establecida la conexión, se obtiene un listado actualizado con las direcciones IP de todos los nodos visibles de la red.

## **Abstract**

The goal of the project is to try to find the solution to the problem of mobile devices interconnection, specifically of the Nokia 810 PDAs, in a situation where we can't count on a fix access point. In order to do this, we create a MANET, which is a mobile Ad-hoc network, where the nodes are the devices themselves. Once we have established the connection, we obtain an updated list with the IP addresses of all the visible nodes in the net.